On Target Counting by Sequential Snapshots of Binary Proximity Sensors*

Tongyang Li[1], Yongcai Wang[1,**], Lei Song[1], and Haisheng Tan[2]

[1] IIIS, Tsinghua University, Beijing, China
[2] Jinan University, Guangzhou, China
wangyc@tsinghua.edu.cn

Abstract. Counting and tracking multiple targets by binary proximity sensors (BPS) is known difficult because a BPS in "on" state cannot distinguish how many targets are presenting in its sensing range. Existing approaches investigated target counting by utilizing joint readings of a network of BPSs, called a snapshot [2,11]. A recent work [14] presented a snapshot-based target counting lower bound. But counting by individual snapshot has not fully utilized the information between the sequential readings of BPSs. This paper exploits the spatial and temporal dependency introduced by a sequence of snapshots to improve the counting bounds and resolution. In particular, a dynamic counting scheme which considers the dependency among the snapshots were developed. It leads to a dynamic lower bound and a dynamic upper bound respectively. Based on them, an improved precisely counting condition was presented. Simulations were conducted to verify the improved counting limits, which showed the improvements than the snapshot-based methods.

1 Introduction

Binary proximity sensors (BPS) is an extracted model for a large category of sensors, such as infrared, ultrasound, microwave, and magnetic sensors. It has an extremely simple sensing model, which outputs a single bit "1" when one or more mobile targets are in its sensing range and "0" otherwise. A BPS sensor cannot distinguish the targets, decides how many distinct targets are presenting in its range, nor judges the targets' moving directions.

Despite of the very limited information provided by one BPS, prior works [11][9] showed the feasibility to track a single target using a collaborative network of BPSs. In [9], the authors showed that if only one target was presenting, the worst case location error is bounded by $\Omega(\frac{1}{\rho R^{d-1}})$, where ρ is the sensor density, R is the sensing range, and d is the dimension of the space. However, significant difficulties are

* This work was supported by in part by National Natural Science Foundation of China Grant 61202360, 61073174, 61033001, 61061130540, the Hi-Tech research and Development Program of China Grant 2006AA10Z216, and the National Basic Research Program of China Grant 2011CBA00300, 2011C-BA00302.

** Corresponding author.

T. Abdelzaher et al. (Eds.): EWSN 2015, LNCS 8965, pp. 19–34, 2015.

encountered for tracking multiple targets because each "on" sensor cannot distinguish how many targets are presenting in its sensing range. Therefore, a fundamental challenge is to count the number of targets precisely.

Existing approach investigated the target counting problem by exploiting individual snapshot captured from a network of BPS. We call such case *static counting*. In [11], Singh et al. presented that accurate target counting could be achieved by a snapshot if the targets are separated by at least $4R$, where R is a sensor's sensing radius. Recent work in [14] presented a lower bound of static counting, which stated that the number of presenting targets equals to the minimum clique partition of the UDG formed by the "on" sensors. However, *static counting* has not fully utilized the information provided by the sensors' reading sequences. In this paper, we show that the upper and lower bounds on target counting can be further improved by exploiting the temporal, spatial dependencies between the sequential snapshots.

Different from the existing approaches, we take the *sequential events reported from sensors during a period* as the problem's input. In this case, a *dynamic counting* technique to infer the lower bound of the target number was designed. We showed theoretically and numerically that the lower bound given by dynamically counting can effectively improve the existing lower bound in static counting. For estimating the upper bound of the number of targets, we firstly propose a packing-based upper bound for snapshot cases under an assumption of minimum pair-wise separation distance between targets. Later on, a dynamic counting algorithm is designed to improve the static upper bound, whose effectiveness is also verified by simulations.

Furthermore, the condition for precisely target counting is discussed in our work. In [12], J. Singh et al. proved that at least $4R$ pairwise separation among targets was required for precisely static counting. In this work, by the upper bound and lower bound obtained from dynamic counting, a new separation distance for precise counting was derived, which reduced the $4R$ separation requirement by approximately $\frac{R}{4}$. It shows that dynamic counting can relax the pair-wise separation condition for precisely target counting.

The rest of the paper is organized as follows. Section 2 presents the problem model and the most related works. Section 3 and Section 4 present the lower bound and the upper bound of the target number by dynamic counting, respectively. In Section 5, the condition for precise target counting is discussed. Section 6 provides simulation results which correspond to our algorithm proposed in Section 3 and Section 4. The paper is concluded in Section 7 with discussion of future directions.

2 Problem Model and Background

2.1 Preliminaries

We consider N binary sensors which are deployed in 2-D area of interest (AOI). Each sensor detects objects within its sensing radius R, and generates one bit of information: "1" for presence of targets and "0" for absence. We assume

that the binary sensing is ideal, noiseless, and provides no other information about the location, speed, or direction information of the targets. All sensors are assumed timely synchronized and their locations are assumed calibrated in an initialization process [8]. We also assume that the AOI is fully covered by the sensing areas of BPSs. Sensors work collaboratively to track the targets. Since the targets move continuously in the AOI, the binary readings of a BPS are efficiently encoded by the time intervals between the BPS's "on" and "off" events. Each BPS reports "1" or "0" when a corresponding transition between 1 and 0 happens, which is enough for a centralized processor to interpolate the real-time states of all sensors at any snapshot. We assume that all "on", "off" events are successively collected via some supporting routing and MAC protocols. At the server side, it receives the sequential events reported from the BPS sensors and reconstructs the sensing snapshots at each event. Each snapshot is a length-M binary vector $\mathbf{S}_t \in [0,1]^M$ at time t. Although the presented techniques are applicable for sensors with non-ideal sensing models, such as non-regular sensible region, we focus our analysis on the disk-shape ideal sensing model.

Patch-based Location Description: Traditionally, the underlying locations of targets are described by "patches" formed by the sensing regions of BPSs. M BPSs can partition the AOI to at most $L \leq M^2 - M + 2$ patches [10,9]. Each patch indicates a region which is covered by the same set of sensors and each patch is coded by a length-M vector based on the coverage situations of M sensors. For example, in a network of four sensors, a patch with code "1100" means the patch is covered by the first two sensors, but is not covered by the other two sensors.

Arc-based Location Description: By taking the event time into consideration, targets' locations can be further narrowed down to *arcs*. When a sensor reports a state transition event, the target that triggers this event must be presenting on the edge of this sensor's sensing region. By jointly considering the states of surrounding sensors, we can infer the target's location to be on an *arc* between two patches whose state change. Each arc can be uniquely encoded by the codes of two neighboring patches. E.g., the arc between "100" and "110" can be coded by "100110". Fig.1 shows an example trace of a target represented by patch sequence and arc-time sequence respectively. The arc representation can specify the location of the target at a given time. We will show in following sections that

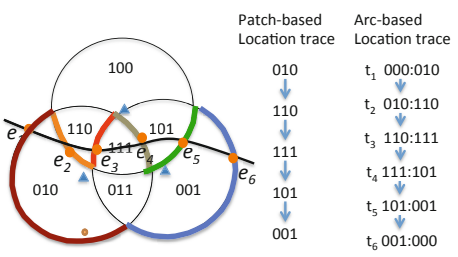


Fig. 1. Location traces of a target represented by patch index and time-arc index respectively

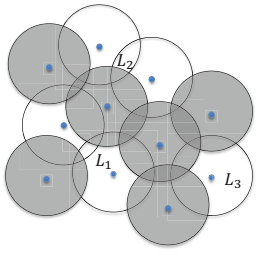


Fig. 2. A snapshot, where feasible areas are partitioned into three isolated islands

Table 1. Notation List

Notation	Meaning
R	Radius of each sensor
M	Number of sensors in the area of interest (AOI)
N	Number of snapshots given in input
v	Total number of isolated feasible islands
L_i	The ith isolated feasible island
$P(s)$	Coverage area of sensor s
f_a	Feasible Crossing Arc (at a certain time t_a)
$T = \{t_1, \cdots, t_k\}$	Set of time corresponding to given snapshots
U	All $t \in T$ whose snapshot triggers on a previous "off" sensor
D	All $t \in T$ whose snapshot triggers off a previous "on" sensor
S_{t_k}	Snapshot at time t_k
G_k	Patch graph at time t_k

this property of arc representation can help to improve the counting resolution by dynamic counting.

Problem specification: Under above system model, we consider multiple target counting problem by sequential snapshots. Each snapshot at t is the captured states of M BPSs when some sensor in the region reports a change. That is, a snapshot at time $t \in T := \{t | \mathbf{S}_t \neq \mathbf{S}_{t-\epsilon}$ for sufficiently small $\epsilon > 0\}$. *The problem input* is a sequence of N snapshots $\{\mathbf{S}_{t_1}, \mathbf{S}_{t_2}, \cdots, \mathbf{S}_{t_N}\}$. Without loss of generality, we assume $t_1 < t_2 < \cdots < t_N$. We also assume the number of targets participating in the AOI will not change during t_1 to t_N and each target's moving speed is upper bounded by V_{max}. *The problem output* is a lower bound and an upper bound of the number of targets.

Notation list: Notations used in our paper are listed in Table 1.

2.2 Background

Target counting problem by one snapshot has been investigated intensively in the literature. A notable concept presented in [11,14] is the *feasible area*. Given a snapshot, the feasible area where targets may present can be determined by $\mathbf{F} = P(\mathbf{A}) - P(\mathbf{A}) \cap P(\mathbf{E})$, where $P(\mathbf{A})$ is the coverage area of the "on" sensors and $P(\mathbf{E})$ is the coverage area of the "off" sensors. An example of the feasible area is shown in Fig.2, in which the sensing regions of "on" sensors are in white and sensing regions of "off" sensors are in grey. In the figure, the feasible area is partitioned into three feasible islands L_1, L_2, L_3, which are called *isolated feasible islands*.

Existing Lower Bounds. For point model of targets, the target number in the feasible area have no upper bound. Estimating the lower bound of target number is the foundation of target counting.

• In [11], a lower bound is given by Singh et al. for counting targets moving in one dimensional space, i.e., on a line. In their method, if "on" sensors can be partitioned into at most X *positively independent sets*, where the positively

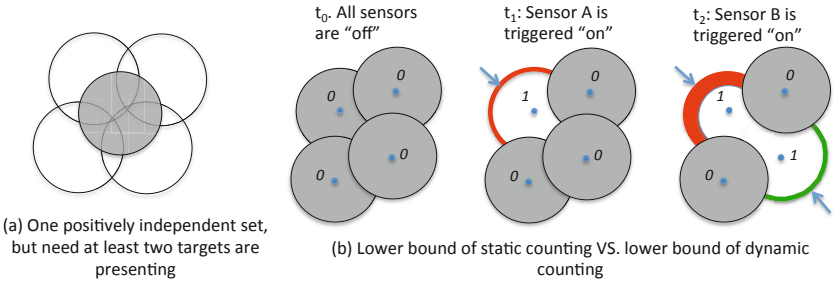


Fig. 3. Examples to illustrate and compare different lower bounds for target counting

independent sensors are "on" sensors whose sensing regions do not overlap, or separated by at least one "off" sensors, Theorem 4 in [11] stated that the number of targets is not less than the cardinarity of X, i.e., $|X|$.

• A recent work in [14] investigated the target number lower bound in 2-D space. They showed that the lower bound given in [11] was conservative in 2-D space. A unit disc graph (UDG) model was proposed to model the structure of the feasible area, based on which an improved lower bound was given. It equals to $\sum_{i=1}^{v} c_i$, where c_i is the minimum number of cliques partitioning the UDG of the ith feasible isolated island.

• In this paper, we show that the lower bound in [14] can be further improved if the temporal and spatial dependences between snapshots are taken into consideration. The basic intuition is shown in Fig.3, which compares the lower bounds mentioned above. In Fig.3a), we can see the four "on" sensors are in one positively dependent set, so that the lower bound given by [11] will be *one*. But the lower bound given by [14] will be *two* because the UDG formed by the "on" sensors has at least two cliques. So the lower bound in [14] is more accurate in 2-D space than that in [11], but it can be further improved. As illustrated in Fig.3(b), at t_2, from the UDG structure, the lower bound given by [14] will be *one*. But by considering the event sequence from t_0 to t_2 and the limited moving speed of the target, we can judge that the target triggers the sensor A at t_1 cannot trigger the sensor B at t_2. Consequently, the lower bound of the target number should be *two* in dynamic counting.

Other Related Works. Most other related works focused on the multiple target tracking algorithms. To deal with the difficulty of multiple targets, Busnel et al. [2][1] investigated the trajectory identification properties. They converted the BPS network into a state graph and presented trajectory identifiable and unidentifiable properties on the state graph. In other works, the number of targets were either assumed known or online estimated by the trajectory disaggregation algorithms. FindingHuMo [4] proposed Hidden Markov Model (HMM) to track a known number of targets by a BPS network. MiningTraMo [17] proposed multiple pairs shortest path algorithm based on walking speed variance to infer the most possible trajectories or targets. In [19], compressive sensing based method

was proposed to count and track the multiple targets, when the targets were known to be sparse, i.e., well separated. In [6], a hybrid multiple target tracking scheme was proposed by He et al., which conducted coarse-scale tracking by binary proximate sensors to narrow down search area, and used high-end sensors for fine-grained tracking. In [3], Cao et al. presented collaborative scheme for tracking groups of targets using BMSs. A distributed PIR-based people number counting system in office environment was developed in [16]. Algorithms and systems for indoor locating using ultrasound systems were investigated in [18][20]. Without going into details of target locating and tracking, we focus on the basic properties of multiple target counting by the sequential snapshots of a BPS network.

3 Lower Bound of Target Number by Dynamic Counting

3.1 Preliminary

We firstly investigate lower bound of target number by utilizing a sequence of snapshots captured by BPSs. For convenience, we divide the time set $T = \{t_1, t_2, \cdots, t_N\}$ corresponding to the given snapshots into two sets U and D, namely *up-set* and *down-set*. $t_k \in U, k \in \mathbb{N}$ if and only if an "off" sensor in $\mathbf{S}_{t_{k-1}}$ is triggered on in $\mathbf{S}_{t_k}$, and $t_k \in D, k \in \mathbb{N}$ if and only if an "on" sensor in $\mathbf{S}_{t_{k-1}}$ is triggered off in $\mathbf{S}_{t_k}$. Next, we define *feasible crossing arc* to indicate the possible locations of the targets that trigger a state transition event.

Definition 1 (Feasible Crossing Arc (FCA)). *When a sensor's state change is detected, the feasible crossing arc indicates the arc segments where the targets are traversing to trigger the event without violating the states of other sensors.*

Based on FCA, we propose Theorem 1 to specify the necessary time-space restriction for two events being triggered by the same target. The theorem is based on the fact that a target's moving speed is limited. Therefore, only if the distance between the FCAs of these two events are not beyond the moving scope of the target, can the two events be triggered by the same target.

Theorem 1 (Time-Space Restriction). *If a sensor A is triggered "on" by one target at time t_A and the FCA is f_A; another sensor B is triggered on by the same target at time $t_B \geq t_A$ with FCA f_B, then $\|f_A - f_B\|_2 \leq (t_B - t_A)V_{\max}$, where V_{max} is the maximum moving speed of the target.*

3.2 Dynamic Counting Using the Time-space Restriction

The time-space restriction could improve the lower bound of the number of targets. An example to show the basic idea is illustrated in Fig.4. It shows six snapshots captured from a BPS network. The ground truth happened during this period is that: three targets, in terms of "red", "green", and "orange" are presented as shown in Snapshot 1. At Snapshot 2, the red target moves outside a little bit, and then turns back quickly as shown in Snapshot 3. Then in a very

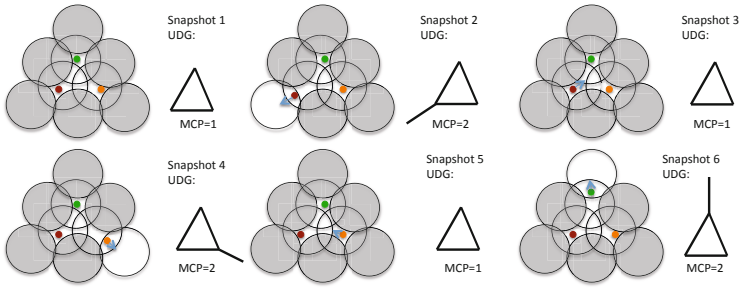


Fig. 4. Improvement by time-space restriction

close snapshot, the orange target moves outside a little bit as shown in Snapshot 4, and then turns back quickly as shown in Snapshot 5. Then in another close by snapshot, the green target moves outside.

By static counting method, the UDGs besides each scenario show the static counting result. The number of MCP for UDG from snapshot 1 to snapshot 6 are one, two, one, two, one, two respectively. Therefore, the estimated lower bound of the number of targets during this period, given by static counting is *two*.

However, since the procedure finishes in ephemeral time, by considering the spatial and temporal dependency between the snapshots we know that the sensors which are triggered on in Snapshot 2, 4 and 6 must be triggered by totally different targets due to the limited speed of the targets. Therefore, in Snapshot 6, we could deduce that a disparate target other than the two targets triggering events at Snapshot 2 and 4 must be presenting. Therefore, the lower bound of the target number is *three* by utilizing time-space restriction.

Based on the idea above, we develop a dynamic counting method to estimate the target lower bound more precisely. To initialize the algorithm, the *beginning patch graph* G_0 is built based on the UDG model in static counting. *Count* is defined as the estimated lower bound of the target number, which is set to $\mathrm{MCP}(G_0)$ initially. After that, a loop runs from the first snapshot to the last snapshot in order to construct patch graphs dynamically. More specifically, for a snapshot at time $t_k \in U$, assume the sensor l is triggered from "off" to "on". In this case, we will firstly construct all edges between sensor l and other intersected "on" sensors whose common intersection is not fully covered by the regions of the "off" sensors. After that, we examine all these edges: if an edge violates the time-space restriction, the edge will be deleted.

For a snapshot at time $t_k \in D$, we need to delete the vertex of the sensor from the UDG, which is just turned from "on" to "off" in the graph, and delete all its corresponding edges. In addition, it is a necessity to delete the edges whose corresponding intersection area is fully covered by this newly "off" sensor.

After finishing each loop, we calculate the MCP of the new patch graph. *count* would be updated if this MCP is larger than the previous *count*. The algorithm ends after looking at all snapshots in time sequential order. The whole procedure is named as dynamic counting of targets, and we develop Algorithm 1 for this method. The dynamic counting algorithm leads to Theorem 2.

Algorithm 1. Lower Bound Dynamic Counting Algorithm

Input: Set $T = \{t_1, t_2, \cdots, t_N\} = U \cup D$ (up-set and down-set); $\mathbf{S}_{t_k}, \mathbf{A}_{t_k}, \mathbf{E}_{t_k}, \forall 0 \leq k \leq N$; Patch graph G_0 of time t_0;
Output: Lower bound of the number of targets: *count*;

```
1  Initialize count ←MCP(G_0);
2  for k ← 1 to N do
3      if t_k ∈ U then
4          Define l to be the sensor which is "off" in S_{t_{k-1}} but "on" in S_{t_k}; F_l to
           be the feasible crossing arc of sensor l;
5          Define G_k = G_{k-1} ∪ {l};
6          for i ← 1 to M,i ≠ l do
7              Define F_i to be the feasible crossing arc of sensor i;
8              if D_{i,l} ≤ 0 AND sensor i and sensor l have intersecting region and
               the intersected region is not fully convered by the regions of the "off"
               sensors then
9                  add an edge in G_k between i and l
       if t_k ∈ D then
10         Define l to be the sensor which is "on" in S_{t_{k-1}} but "off" in S_{t_k};
11         Define G_k = G_{k-1}/{l} by deleting vertex l and its corresponding edges;
12         if sensor i, j, l have intersecting region pairwise and the intersected
           region of sensor i and j is fully convered sensor l then
13             delete the edge in G_k between i and j
14     count ← max{count, MCP(G_k)};
15 return count;
```

Theorem 2 (Lower Bound of the Target Number). *Let N_L be the real number of targets that matches the sequential snapshots of sensors. Then the return value 'count' of Algorithm 1 must not be larger than N_L, i.e., $N_L \geq count$.*
Proof. Based on the discussion of Algorithm 1, it is clear that G_k constitutes the snapshot at time t_k. In order to prove the theorem, we suffice to show that $N \geq \text{MCP}(G_k)$ for each k since *count* is the maximum of all $\text{MCP}(G_k)$.

Let us assume the contrary that $N < \text{MCP}(G_k)$ for a certain k. As a result, we could re-partition all "on" sensors of the UDG at time t_k into $\text{MCP}(G_k) - 1$ groups such that each group of sensors has a common intersection while each pair of sensors does not violate the time-space restriction. This partition is equivalent to a clique partition of G_k with $\text{MCP}(G_k) - 1$ cliques, but this contradicts with the fact that $\text{MCP}(G_k)$ is a minimum clique partition of G_k.

Our algorithm also has a reasonable time complexity. By [5], MCP of a patch graph could be calculated by a polynomial time approximation scheme (PTAS) with $(1 + \epsilon)$-approximation and time complexity $O(M^{O(1/\epsilon^2)})$ where M is the number of sensors in the area of interest (AOI). In addition, line 2 to line 13 in the algorithm could be done in $O(M)$ time in each cycle. Therefore, Algorithm 1 is also a PTAS with $(1+\epsilon)$-approximation and has time complexity $O(N \cdot M^{O(1/\epsilon^2)})$ where N is the number of snapshots.

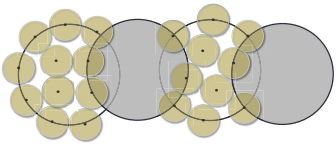


Fig. 5. Example of calculating the upper bound of target number

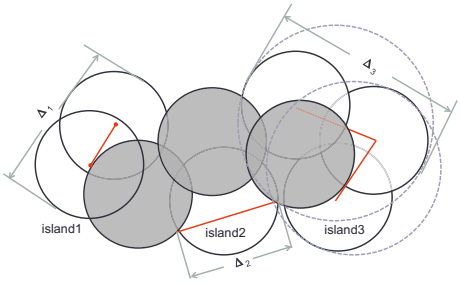


Fig. 6. Example of minimum separation distance

4 Upper Bound by Dynamic Counting

In real applications, the physical targets, such as humans, animals, vehicles, are generally not arbitrarily close to each other. In this section, we assume a *minimum separation distance* $r > 0$ between each pair of targets.

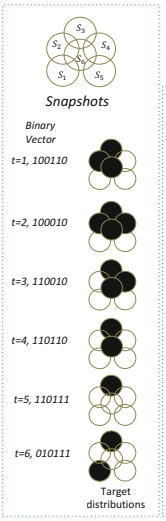
4.1 Static Counting

Based on the assumption of minimum separation distance, we can estimate the upper bound of target number by modeling it as a packing problem, which is a classical geometric optimization problem in mathematics that attempts to pack objects together into containers. The goal is to pack the containers as densely as possible using the objects. In 1910, Thue [15] established a theorem for the density of circle packing into a connected surface:

Theorem 3. *Assume a set of at least two circles with radius r are packed into a connected surface. Denote the sum of area of all small circles with radius r to be S' and denote the area of the surface to be S, respectively. Then $\frac{S'}{S} \leq \frac{\pi}{\sqrt{12}}$.*

In our problem, we treat the areas of each feasible island as containers, and objects are circles with radius r. Then the upper bound of the number of targets equals to the number of circles that can be packed into the feasible area. In our problem, we only restrict the centers of objects, i.e., the positions of targets cannot exceed the boundary of the container. Therefore, we allow the objects to cover at most distance r beyond the boundary of containers, as an example shown in Fig.5. Given a snapshot as the input, the most number of targets in the feasible area of the snapshot can be estimated.

Theorem 4 (Upper Bound of the Target Number). *Let A_i and C_i be the area and circumference of the ith feasible island respectively. Suppose the minimum separation distance between targets is r, then the number of targets in the feasible area must be smaller than $\sum_{i=1}^{v} \frac{A_i + rC_i}{\sqrt{12}r^2}$, where $\frac{A_i + rC_i}{\sqrt{12}r^2}$ is the upper bound of target number in the ith feasible island.*



Number of isolated islands	Upper bound of island 1	Upper bound of island 2	Upper bound Overall
2	$F(S_1)$	$F(S_4 \cup S_5)$	$F(S_1)+F(S_4 \cup S_5)$
2	$F(S_1)$	$F(S_4)$	$F(S_1)+F(S_4)$
2	$F(S_1)$	$F(S_4)$	$F(S_1)+F(S_4)$
2	$F(S_1)$	$F(S_4)$	$F(S_1)+F(S_4)$
1	$F(S_1)+F(S_4)$	N/A	$F(S_1)+F(S_4)$
1	$F(S_1)+F(S_4)$	N/A	$F(S_1)+F(S_4)$

Fig. 7. Illustration for dynamic upper bound

Using the similar idea of improving lower bound, we introduce a dynamic counting method to utilize snapshot dependences to improve the upper bound of the target number.

4.2 Improvement by Dynamic Counting

For a feasible region S, we define $f(S) := \frac{A+rC}{\sqrt{12}r^2}$ where A is the area of S and C is the circumference of S. By Theorem 4, $f(S)$ is an upper bound of the number of targets in S. Denote the set of feasible islands of the initial snapshot at time t_0 to be $S_{island} = \{S_1, \cdots, S_K\}$.

The basic idea of dynamic counting to improve the upper bound is based on the fixed number of targets in a feasible island which is not connected to other feasible islands during the concerned time. See Fig. 7 as an illustration. This is an instance of six snapshots, in which the first four snapshots are composed of two feasible islands and the last two snapshots are composed of a single feasible island. From snapshot 1 and 2, we know that the number of targets in the first and second feasible island is at most $f(S_1)$ and $f(S_5)$, respectively. Therefore, the best upper bound of the number of targets in this instance is $f(S_1) + f(S_5)$. However, if we only consider the last snapshot, the upper bound obtained is $f(S_2 \cup S_4 \cup S_5 \cup S_6)$, which could be much worse than the bound given from the dynamic view. Using this inspiration, we develop Algorithm 2 to improve the upper bound of the number of targets. Basically, when the number of feasible islands is not changed in an interval of snapshots, the number of targets in each feasible island is bounded by its smallest region in this interval. When a feasible island breaks up into several islands or some feasible islands are combined into a single feasible island, the set of feasible islands and the upper bound of the number of targets in each feasible island are both reassigned.

Algorithm 2. Dynamic Upper Bound Counting Algorithm

Input: Set $T = \{t_1, t_2, \cdots, t_N\} = U \cup D$ (up-set and down-set);
$\mathbf{S}_{t_k}, \mathbf{A}_{t_k}, \mathbf{E}_{t_k}, \forall 0 \leq k \leq N$; Feasible islands $S_{island} = \{S_1, \cdots, S_K\}$ at time t_0;
Output: Upper bound of the number of targets: $count_{upper}$;

1 **for** $i \leftarrow 1$ **to** K **do**
2 $u_i \leftarrow f(S_i)$
3 Initialize $count_{upper} \leftarrow \sum_{i=1}^{k} u_i$;
4 **for** $k \leftarrow 1$ **to** N **do**
5 **if** $t_k \in U$ **then**
6 **Define** l to be the sensor which is "off" in $\mathbf{S}_{t_{k-1}}$ but "on" in $\mathbf{S}_{t_k}$;
7 **if** $P(l) \cap S_{island} = \emptyset$ **then**
8 **Update** $S_{island} \leftarrow S_{island} \cup P(l)$; $u_l \leftarrow f(P(l))$;
9 **else if** $P(l)$ *intersects with at least two islands in* S_{island} **then**
10 **Define** $S' \subseteq S_{island}$ to be islands intersect with $P(l)$;
11 **Define** $S_l := S' \cup P(l)$;
12 **Update** $S_{island} \leftarrow (S_{island}/S') \cup S_l$;
13 **Update** $u_l \leftarrow \sum_{S_i \in S'} f(S_i) + f(P(l))$;
14 **else**
  if $t_k \in D$ **then**
15 **Define** l to be the sensor which is "on" in $\mathbf{S}_{t_{k-1}}$ but "off" in $\mathbf{S}_{t_k}$;
16 **Define** l to be in feasible island S_i;
17 **if** $S_i/P(l)$ *is not connective* **then**
18 **Define** $S_i/P(l)$ to be $m \geq 2$ feasible islands $S_{i1}, \cdots, S_{im}$;
19 **Update** $S_{island} \leftarrow (S_{island}/S_i) \cup \{S_{i1}, \cdots, S_{im}\}$;
20 $u_{ij} \leftarrow f(S_{ij})$ for $\forall 1 \leq j \leq m$;
  else
21 $u_i \leftarrow \min\{u_i, f(S_i/P(l))\}$;
22 $count_{upper} \leftarrow \min\{count_{upper}, \sum_i u_i\}$
23 **return** $count_{upper}$;

In the algorithm, we also divide the time of snapshots into up-set and down-set. For both cases, the isolation and combination of islands are carefully constructed to make the connectivity of all "on" sensors following the truth. In addition, we update each island's upper bound of the number of targets every round. In particular, when an "on" sensor is turned off in a snapshot, we catch up the possibility of decreasing the upper bound of the target number in line 21.

In Algorithm 2, $O(M)$ time suffices from line 1 to line 3 where M is the number of sensors in the AOI. In either $t_k \in U$ or $t_k \in D$, the cycle could be finished in $O(M)$ time. In total, the time complexity of Algorithm 2 is $O(MN)$ where N is the number of snapshots in total.

Based on the lower bound and the upper bound of target number, we can investigate a more interesting property of the binary target counting problem, i.e., the minimum separation distance for precisely target counting.

5 Condition for Precisely Target Counting

A more interesting problem we may ask is: under what condition can we always precisely count the number of targets without error. This problem was previously studied by [12], which showed that when the separation distance r between each pair of targets is larger than $4R$, the number of targets can be precisely counted. This traditional requirement of $4R$ separation distance is rather large. What we are interested is that: whether can we find a smaller separation distance $r < 4R$ such that the number of targets can be precisely counted.

Consider the relationship between the upper bound and lower bound of the target number in a feasible island. For the ith feasible island, suppose that the lower bound of the target number is l_i. When the separation distance between each pair of targets is r_i, the following relationship must hold:

$$\frac{A_i + r_i C_i}{\sqrt{12} r_i^2} > N_i \geq l_i$$

The minimally required separation distance for precisely target counting in island i is the minimum value of r_i which restricts the upper bound of target number not larger than one plus the lower bound of the target number, i.e., the minimum value of r_i to keep $l_i + 1 > \frac{A_i + r_i C_i}{\sqrt{12} r_i^2} \geq l_i$. Therefore:

Theorem 5. *The minimum required separation distance between each pair of targets for precisely counting targets is* $\gamma = \max\{\gamma_1, \gamma_2, \cdots, \gamma_v\}$, *where* $\gamma_i = \frac{C_i + \sqrt{C_i^2 + 8\sqrt{3} A_i (l_i + 1)}}{2\sqrt{3}(l_i + 1)}$ *for all* i.

Here γ_i is the solution of the equation $\sqrt{12}(l_i + 1)\frac{\gamma_i}{2}^2 - C_i \frac{\gamma_i}{2} - A_i = 0$, which is the minimum separation distance to make the upper bound of target number equal to the lower bound in island i. Moreover, γ_i could be even smaller if use the upper bound of the target number by dynamic counting.

Theorem 5 reveals that the separation distance required for precisely target counting is different in disparate snapshots and even varies at different locations in a snapshot. If the upper bound is unlimited, the minimum separation distance is the largest diameter of the cliques formed by the positive sensors. An example is shown in Fig.6, we can see the minimum separation distance $\min\{\Delta_1, \Delta_2, \Delta_3\} < 4R$, which shows a better potential of using BPS network for precisely target counting than the traditional results.

6 Evaluation

To verify the counting bound, agents based simulation was conducted based on *PSensorSimulator* platform[13]. Multiple agents, which simulated the mobile targets were programmed to move independently along random paths in the area of interest. The area of interest was a $L \times L$ rectangle area. In the area, M BPS sensors were deployed. We investigated two kinds of sensor deployment. 1) *regular deployment*, as shown in Fig. 8, in which the sensors were deployed in

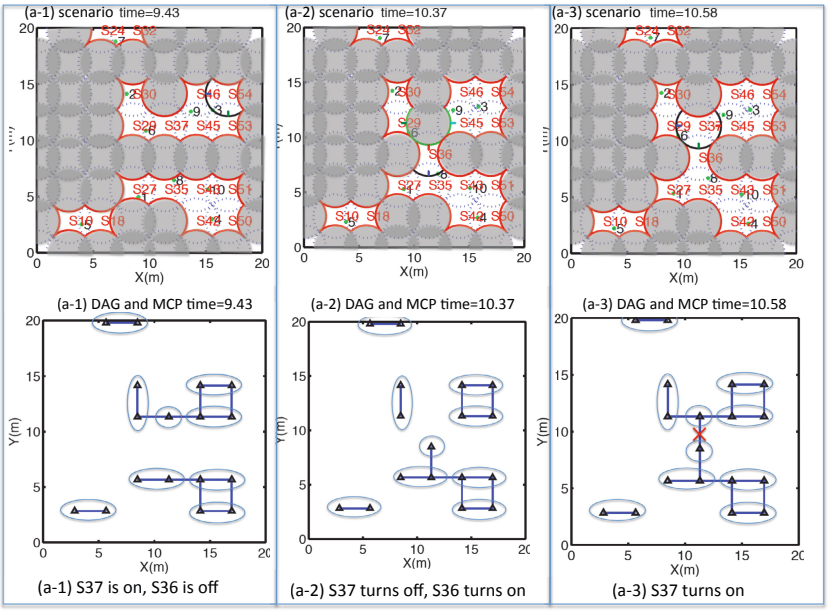


Fig. 8. Dynamic counting can fix boundary-pacing error

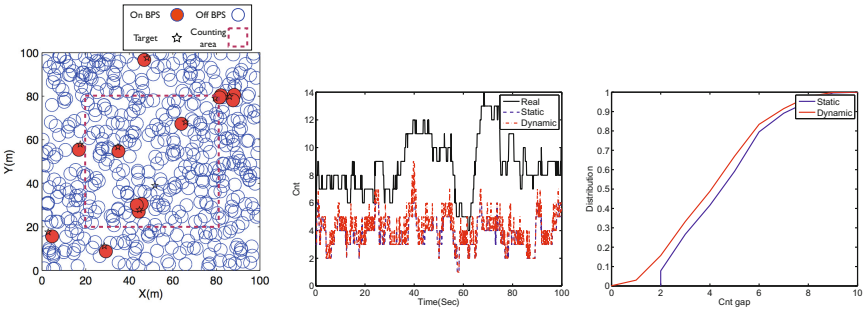
a grid topology, which fully covered the AOI. 2)*random deployment*, in which, enough sensors were deployed randomly in a region and a subregion was selected as the AOI, as shown in Fig.9(a).

For rendering the target tracking scenario, a graphical interface was developed. As shown in Fig.8, the feasible area of targets were rendered by the internal region surrounded by the red arcs. The sensors in the "off" state were in grey with blue dashed lines. The feasible crossing arcs were colored in black if one target was entering the sensor region, and was colored green if the sensor was turned off because of target leaving. The UDG corresponding to the sensor's readings was illustrated in Fig.8, in which the vertex denoted the sensors in the "on" state. The construction of UDG could be referred to [14]. We implemented Algorithm 1 and 2 on the simulation platform to contrast the upper and lower bound with the ground-truth number of targets.

6.1 Evaluation on Lower-bound

As a core unit of Algorithm 1, MCP-calculation routine is called every time the state of a sensor changes. Since MCP-calculation is proved to be NP-Complete problem, we use a PTAS approximation to implement MCP [7] calculation.

Regular deployment: At first, a particular example is shown in Fig.8 to illustrate the effectiveness of the dynamic counting algorithm on improving the counting lower bound. The scenario contains 64 sensors. Only the labels of the "on" sensors are shown, and ten targets are moving in the area. As shown in Fig.8, at 9.43 second in this scenario, the sensor 36 was "off" and sensor 37 was



(a) Setting of random deployed Presence sensor (b) Counting result (c) CDF of counting error

Fig. 9. Dynamic counting vs statistic counting on random deployment

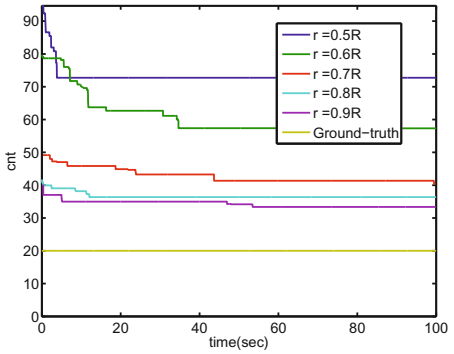


Fig. 10. UpperBound of differen r

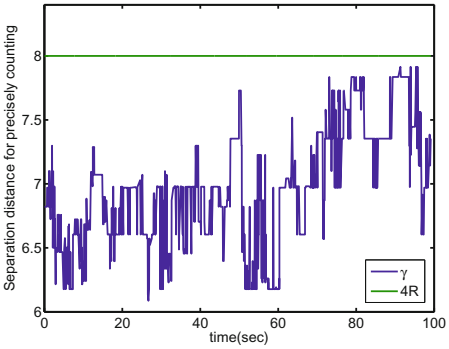


Fig. 11. Separation versus $4R$

"on". The MCP of the UDG of this scenario indicates the lower bound of the target number is 9. At the 10.37 second, sensor 36 turned "on" and sensor 37 turned "off" simultaneously. In this case, the target which left sensor 37 could only enter sensor 29 or sensor 45. After a short duration of 0.21 seconds, at the 10.58 second, the sensor 37 was turned "on" again. Since 0.21 seconds was not enough for the target which triggered sensor 36 to reach the edge of sensor 37 due to time-space restriction, it must be the return of the previous target in sensor 37, and this target is different from the target in sensor 36. Therefore, in the DAG of (a-3), the edge between S36 and S37 was deleted. Deletion of this edge improved the lower bound from 9 to 10, which verified the effectiveness of the dynamic counting for lower bound improvement.

Random Deployment: To further investigate the performance of dynamic counting method, we evaluated the target counting performances when the sensors are randomly deployed. The setting is shown in Fig.9(a), in which sensors are deployed with density of 0.2 per square meter in an 100m*100m area. A subregion in the centric part of the area is selected as the AOI. So that in this evaluation, the number of targets within the AOI may change overtime, therefore, we delete line 14 in algorithm 3 in simulation. The number of targets in AOI given by both

dynamic and static counting algorithm were evaluated. The result in a 100 seconds experiment is shown in Fig.9(b). We can see that the lower bound given by dynamic counting is slightly better than that given by static counting. The CDF curve in Fig.9(c) summarized the performance difference, in which the counting gap indicates the gap to the real number of targets. In conclusion, dynamic counting showed better performance than the static counting.

6.2 Evaluation on Upper-bound

To verify the upper bound of counting, Algorithm 2 was implemented in *PSensorSimulator*. For each isolated island formed by "on" sensors, circumference and area were calculated with numeric method. According to Theorem 4, ratio between predefined separation radius and sensing radius matters. Therefore, we calculated the upper bound with different ratio. As shown in Fig.10, the upper bound could be twice to 4 times to the ground-truth according to different ratios.

6.3 Evaluation on Separation Distance

Theorem 5 gives a non-trivial minimum separation distance γ for precise counting. We compared this γ with classical separation distance $4R$ by simulation. As shown in Fig.11, during 100 seconds experiment, the γ is always below $4R$. This results told us that, introducing dynamic information can improve the separation for precisely counting by about $\frac{R}{4}$.

7 Conclusion

This paper investigated target counting problem by a network of binary proximity sensors. For the lower bound of the target number, we considered the time-space restriction between a sequence of snapshots and proposed a dynamic counting technique which improved the lower bound given by individual snapshot. As for the upper bound of the target number, we showed that if a minimum separation distance between targets was considered, an upper bound could be given by packing theorem. Moreover, a dynamic counting algorithm was proposed to improve this upper bound. At last, by matching the upper bound and lower bound, we investigated the condition for precisely target counting and showed that the minimum separation distance for precisely counting could be $\frac{R}{4}$ smaller than the previously known limit $4R$. In the future work, the dynamic counting method can be exploited to enhance existing multiple target tracking algorithms. Apart from theoretical works, dynamic counting technique can be applied in occupying sensing or enemy detection and tracking.

References

1. Busnel, Y., Querzoni, L., Baldoni, R., Bertier, M., Kermarrec, A.-M.: On the deterministic tracking of moving objects with a binary sensor network. In: Nikoletseas, S.E., Chlebus, B.S., Johnson, D.B., Krishnamachari, B. (eds.) DCOSS 2008. LNCS, vol. 5067, pp. 46–59. Springer, Heidelberg (2008), http://link.springer.com/chapter/10.1007/978-3-540-69170-9_4

2. Busnel, Y., Querzoni, L., Baldoni, R., Bertier, M., Kermarrec, A.-M.: Analysis of deterministic tracking of multiple objects using a binary sensor network. ACM Trans. Sen. Netw. 8(1), 8:1–8:27 (2011)
3. Cao, D., Jin, B., Das, S.K., Cao, J.: On collaborative tracking of a target group using binary proximity sensors. Journal of Parallel and Distributed Computing 70(8), 825–838 (2010)
4. De, D., Song, W.-Z., Xu, M., Wang, C.-L., Cook, D., Huo, X.: FindingHuMo: real-time tracking of motion trajectories from anonymous binary sensing in smart environments. In: ICDCS, pp. 163–172 (2012)
5. Dumitrescu, A., Pach, J.: Minimum clique partition in unit disk graphs. Graphs and Combinatorics 27(3), 399–411 (2011)
6. He, T., Bisdikian, C., Kaplan, L., Wei, W., Towsley, D.: Multi-target tracking using proximity sensors. In: MILCOM, pp. 1777–1782 (2010)
7. Humyn, A.: Maximal Clique (May 2008)
8. Liu, H., Darabi, H., Banerjee, P., Liu, J.: Survey of wireless indoor positioning techniques and systems. Trans. Sys. Man Cyber Part C 37(6), 1067–1080 (2007)
9. Shrivastava, N., Madhow, R.M.U., Suri, S.: Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms. In: SenSys 2006, pp. 251–264. ACM, New York (2006)
10. Shrivastava, N., Mudumbai, R., Madhow, U., Suri, S.: Target tracking with binary proximity sensors. ACM Trans. Sen. Netw. 5(4), 30:1–30:33 (2009)
11. Singh, J., Madhow, U., Kumar, R., Suri, S., Cagley, R.: Tracking multiple targets using binary proximity sensors. In: IPSN 2007, pp. 529–538. ACM, New York (2007)
12. Singh, J., Madhow, U., Kumar, R., Suri, S., Cagley, R.: Tracking multiple targets using binary proximity sensors. In: IPSN 2007. ACM (April 2007)
13. Song, L.: PSensorSimulator, `https://bitbucket.org/leisong03/PSensorSimulator`
14. Song, L., Wang, Y.: Multiple target counting and tracking using binary proximity sensors: Bounds, coloring, and filter. In: Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2014, pp. 397–406. ACM, New York (2014)
15. Thue, A.: Über die dichteste Zusammenstellung von kongruenten Kreisen in einer Ebene. na (1910)
16. Wahl, F., Milenkovic, M., Amft, O.: A distributed PIR-based approach for estimating people count in office environments. In: CSE 2012, pp. 640–647 (2012)
17. Wang, C., De, D., Song, W.-Z.: Trajectory mining from anonymous binary motion sensors in smart environment. Knowledge-Based Systems 37, 346–356 (2013)
18. Wang, Y., Song, L.: An algorithmic and systematic approach for improving robustness of TOA-based localization. In: 2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), pp. 2066–2073 (November 2013)
19. Zhang, B., Cheng, X., Zhang, N., Cui, Y., Li, Y., Liang, Q.: Sparse target counting and localization in sensor networks based on compressive sensing. In: 2011 Proceedings IEEE INFOCOM, pp. 2255–2263 (April 2011)
20. Zhao, J., Wang, Y.: Autonomous ultrasonic indoor tracking system. In: International Symposium on Parallel and Distributed Processing with Applications, ISPA 2008, pp. 532–539 (December 2008)